Unit-6
Application Layer

PREPARED BY: SUSHANT BHATTARAI

Introduction

- The application layer provides services to the user.
- Communication is provided using a logical connection, which means that the two application layers assume that there is an imaginary direct connection through which they can send and receive messages
- Function of Application Layer
  - File transfer
  - Mail services
  - Delivery Services

DNS

- The Domain Name System (DNS) is a central part of the Internet, providing a way to match names (a website you're seeking) to numbers (the address for the website).
- Anything connected to the Internet -laptops, tablets, mobile phones, websites -has an Internet Protocol (IP) address made up of numbers. Your favorite website might have an IP address like 64.202.189.170, but this is obviously not easy to remember. However a domain name such as bestdomainnameever.comis something people can recognize and remember.
- DNS syncs up domain names with IP addresses enabling humans to use memorable domain names while computers on the Internet can use IP addresses

Namespace

- The names must be unique because the addresses are unique. A namespace that maps each address to a unique name can be organized in two ways:
- Flat namespace
  - In a flat name space, a name is assigned to an address. A name in this space is a sequence of characters without structure.
  - The names mayor may not have a common section; if they do, it has no meaning.
  - The main disadvantage of a fiat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Namespace

- Hierarchical namespace
  - In a hierarchical name space, each name is made of several parts. The first part can define the nature of the organization, the second part can define the name of an organization, the third part can define departments in the organization, and so on.
  - In this case, the authority to assign and control the name spaces can be decentralized.
  - A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.

Domain Namespace

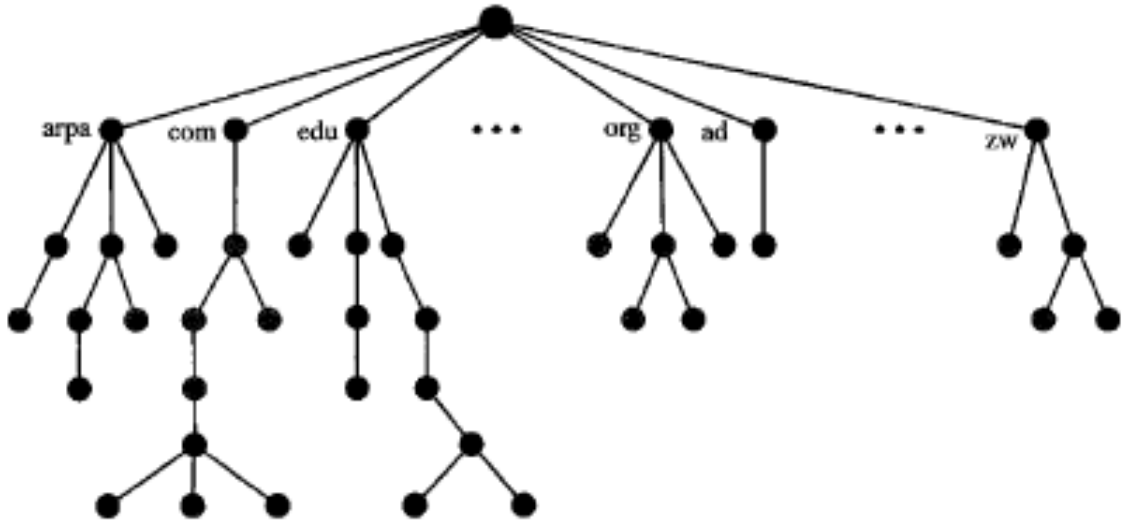
- To have a hierarchical name space, a domain name space was designed.
- In this design the names are defined in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels: level 0 (root) to level 127
- **Label**
  - Each node in the tree has a label, which is a string with a maximum of 63 characters.
  - The root label is a null string (empty string).
  - DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Namespace

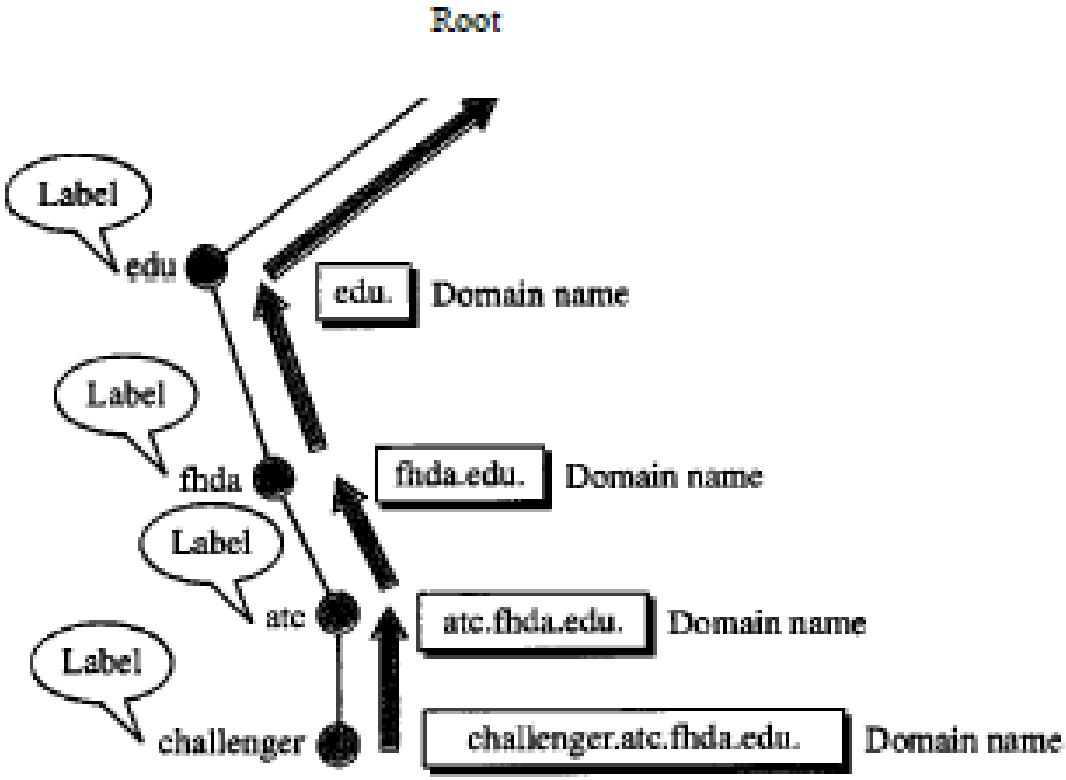
- Domain Name
- Each node in the tree has a domain name. A full domain name is a sequence of labels separated by dots (.).
- The domain names are always read from the node up to the root.
- The last label is the label of the root (null).
- This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

Domain Namespace

e 25.2 *Domain name space*



Domain names and labels



DNS Query

- There are three types of queries in the DNS system:
- Recursive Query
  - In a recursive query, a DNS client provides a hostname, and the DNS Resolver "must" provide an answer—it responds with either a relevant resource record, or an error message if it can't be found.
  - The resolver starts a recursive query process, starting from the DNS Root Server, until it finds the Authoritative Name Server (for more on Authoritative Name Servers see DNS Server Types below) that holds the IP address and other information for the requested hostname.
- Iterative Query
  - In an iterative query, a DNS client provides a hostname, and the DNS Resolver returns the best answer it can.
  - If the DNS resolver has the relevant DNS records in its cache, it returns them. If not, it refers the DNS client to the Root Server, or another Authoritative Name Server which is nearest to the required DNS zone.
  - The DNS client must then repeat the query directly against the DNS server it was referred to.

DNS Query

- Non-Recursive Query
  - A non-recursive query is a query in which the DNS Resolver already knows the answer. It either immediately returns a DNS record because it already stores it in local cache, or queries a DNS Name Server which is authoritative for the record, meaning it definitely holds the correct IP for that hostname.
  - In both cases, there is no need for additional rounds of queries (like in recursive or iterative queries). Rather, a response is immediately returned to the client

Services Provided by DNS

- Translating hostnames to IP addresses
- Host Aliasing
- Mail Server Aliasing
- Load Distribution

How DNS works

- There are 4 DNS servers involved in loading a webpage:
- DNS recursor-
  - The recursor can be thought of as a librarian who is asked to go find a particular book somewhere in a library.
  - The DNS recursor is a server designed to receive queries from client machines through applications such as web browsers.
  - Typically the recursor is then responsible for making additional requests in order to satisfy the client's DNS query.
- **Root nameserver**-
  - The root serveris the first step in translating (resolving) human readable host names into IP addresses.
  - It can be thought of like an index in a library that points to different racks of books –
  - typically it serves as a reference to other more specific locations.

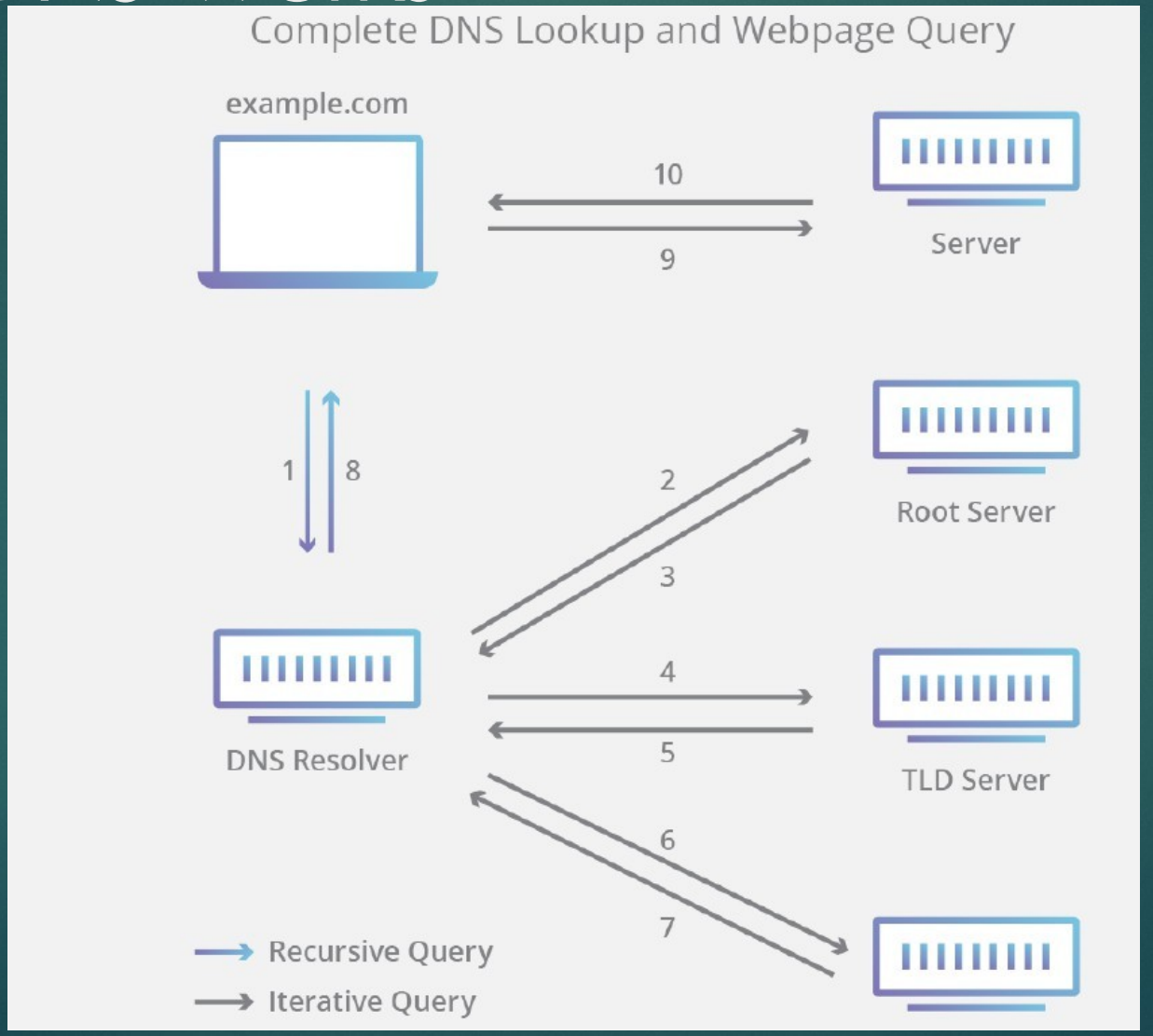
How DNS works

- TLD nameserver-
  - The top level domain server (TLD) can be thought of as a specific rack of books in a library.
  - This name server is the next step in the search for a specific IP address, and
  - it hosts the last portion of a hostname (In example.com, the TLD server is "com").
- Authoritative nameserver-
  - This final name server can be thought of as a dictionary on a rack of books, in which a specific name can be translated into its definition.
  - The authoritative nameserver is the last stop in the name server query.
  - If the authoritative name server has access to the requested record, i

How DNS works

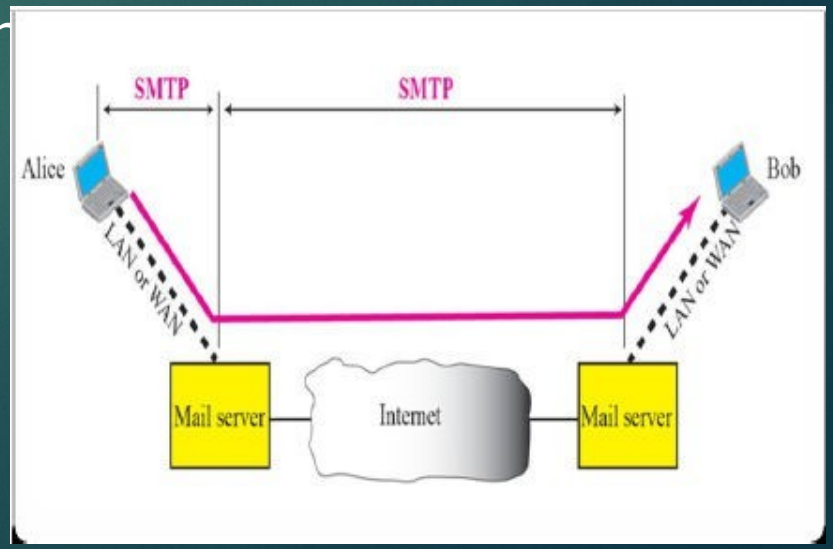
- **Step1:**the client types www.example.com in his browser
- **Step2:**the operating system looks at/etc/host file,first for the ip address of www.example.com(this can be changed from/etc/nsswitch),then looks/etc/resolv.conf for the DNS server IP for that machine
- **Step3:**the dns server will search its database for the name www.example.com , if it finds it will give that back, if not it will query the **rootserver**(.)for the information.
- **Step4:**root server will return are feral to the .com TLD(top level domain)nameserver(these TLD name servers knows the address of name servers of all SLD's).In our case we searched for www.example.com so root server will give us referral to.com TLD servers.
- If it was www.example.net then root server will give, .net TLD servers refferal.
- **Step5:**Now One of the TLD servers of .com will give us the referral to the DNS server responsible for example.com domain.
- **Step6:**the dns server for example.com domain will now give the client the ip address of www host(www is the host name.)

HOW DNS works



SMTP

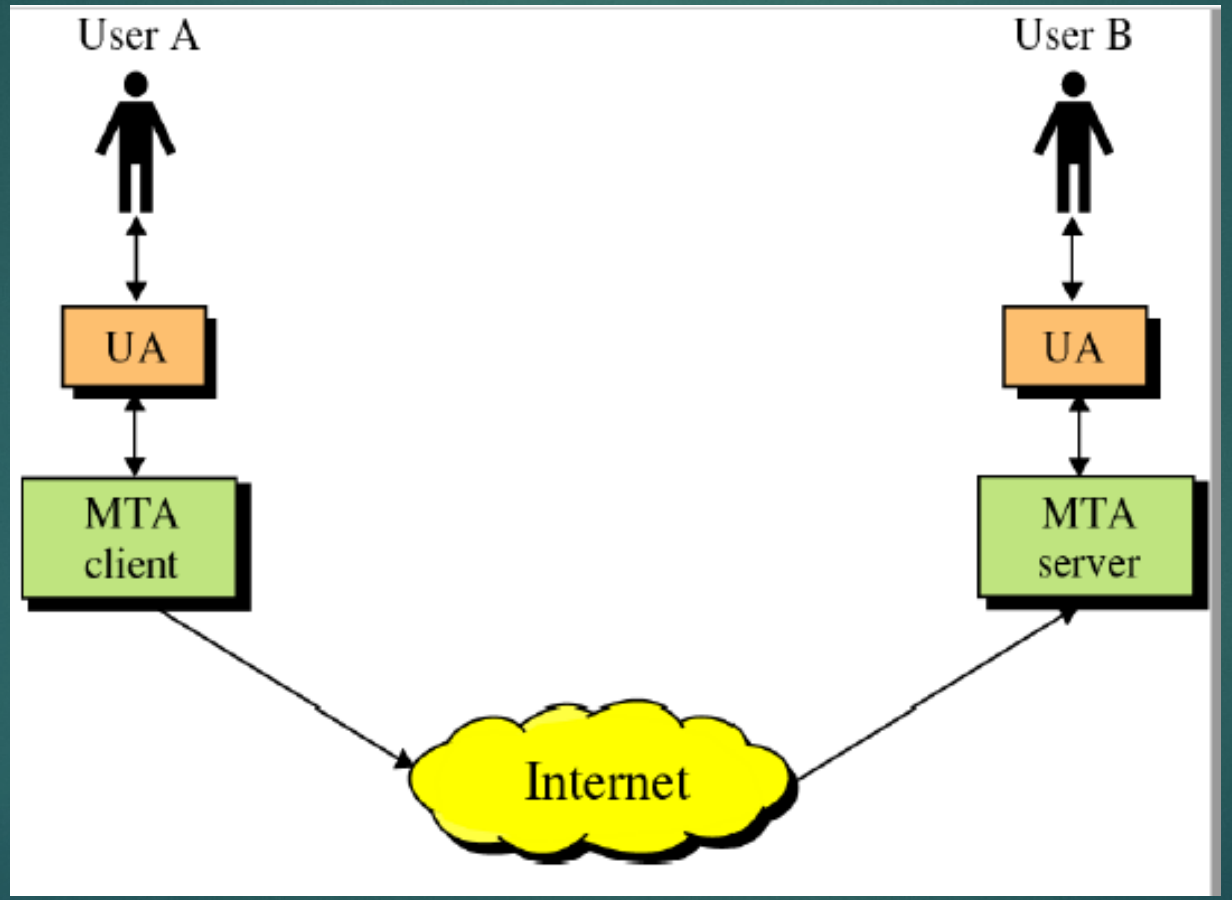
- Most of the internet systems use SMTP as a method to transfer mail from one user to another.
- SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those mails at the receiver's side. It works with post office protocol (POP).
- It is a TCP/IP protocol that specifies how computers exchange electronic mail.
- SMTP is used two times , between the sender & the sender's mail server and
between the two servers.
- SMTP simply define how command & response must be back & for Each
network is free to choose a software package for implementation



SMTP

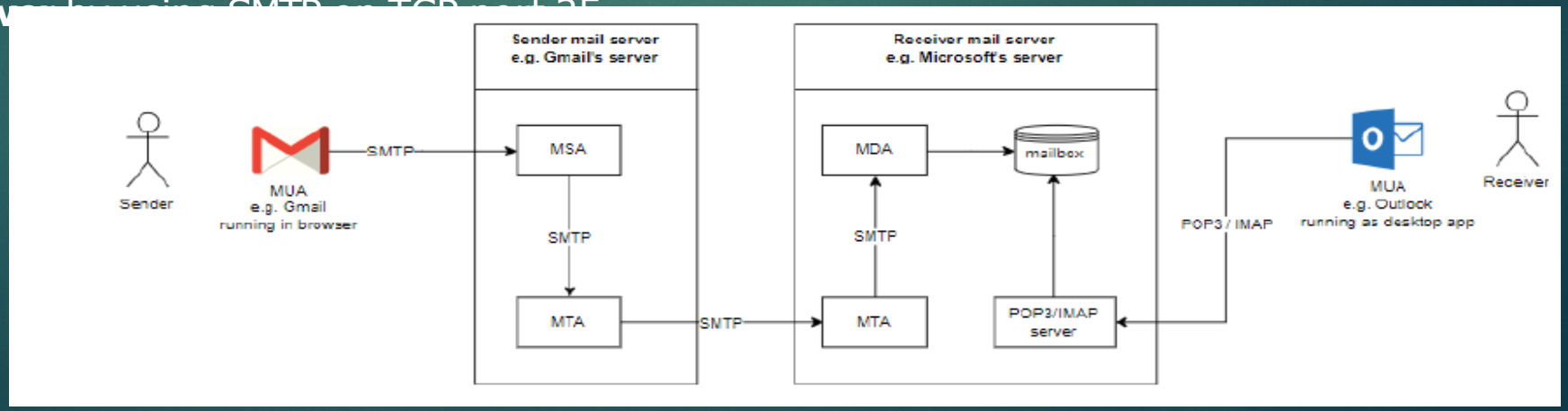
- SMTP clients and servers have two main components:
- User Agents (UA):
- It prepares a message and encloses in an envelope.
- Mail Transfer Agents(MTA):
- SMTP stands for Simple Mail Protocol. The actual mail transfer is done through message transfer agents.
- The formal protocol that defines the MTA client & server in the Internet is called the simple mail transfer protocol
- To send mail, a system must have the client MTA & to receive mail, a system must have a server MTA.
- The MTA maintains a small queue of mails so that it can schedule repeat delivery of mail in case the receiver is not available. The MTA delivers the mail to the mailboxes and the information can later be downloaded by the user agents.

SMTP



Working of SMTP

- Composition of Mail:
  - A user sends an e-mail by composing an electronic mail message **using a Mail User Agent** (MUA). Mail User Agent is a program which is used to send and receive mail.
  - The message contains two parts: body and header.
  - The **body** is the main part of the message while the header includes information such as the sender and recipient address.
  - The **header** also includes descriptive information such as the subject of the message
- **Submission of Mail:**
  - After composing an email, the mail **client then submits the completed e-mail to the SMTP server** by using SMTP on TCP port 25.

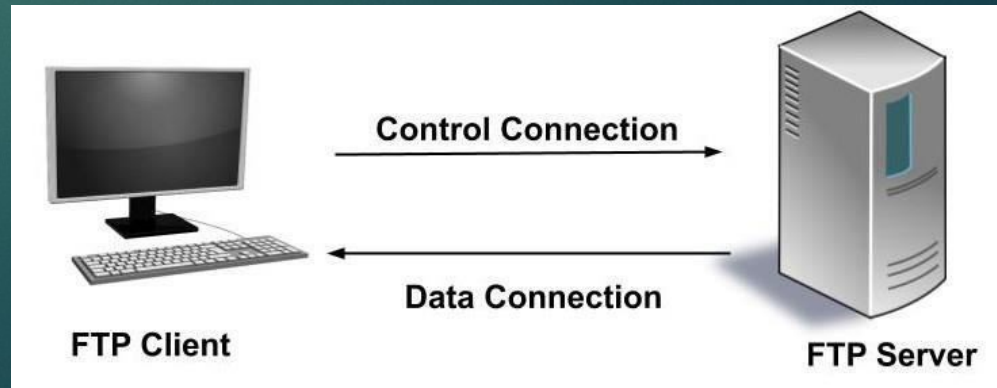


Working of SMTP

- Delivery of Mail:
  - **E-mail addresses contain two parts: username of the recipient and domain name.** For example, vivek@gmail.com, where "vivek" is the username of the recipient and "gmail.com" is the domain name.
  - If the domain name of the recipient's email address is different from the sender's domain name, then MSA (mail submission agent) will send the mail to the Mail Transfer Agent (MTA).
  - To relay the email, the MTA will find the target domain. It checks the MX record from Domain Name System to obtain the target domain. The MX record contains the domain name and IP address of the recipient's domain. Once the record is located, MTA connects to the exchange server to relay the
- **Receipt and Processing of Mail:**
  - Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the e-mail where it waits for the user to retrieve it.
- Access and Retrieval of Mail:
  - The stored email in MDA(mail delivery agent) can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.

FTP(File Transfer Protocol)

- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- FTP is aclient-serverprotocol that relies on two communications channels between client and server:
- **command channel** for controlling the conversation and
- **data channel** for transmitting file content.
- **Control Connection:**The FTP client, for example, FileZilla or FileZilla Pro sends a connection request usually to server port number 21. This is the control connection. It is used for sending and receiving commands and responses. Typically a user needs to**log on**to the FTP server for establishing the connection but there are some servers that make all their content available without login. These servers are known as**anonymous FTP.**



FTP

- **Data Connection:**For transferring the files and folder we use a separate connection called data connection.
- FTP sessions work in passive or active modes.
- •In **active mode**, after **a client initiates a session via a command channel request,** the server initiates a data connection back to the client and begins transferring data.
- •In passive mode, the server instead uses the command channel to send the client the information it needs to **open a data channel**. Because passive mode has the client initiating all connections, it works well across firewalls and Network Address Translation (NAT) gateways.

FTP

- **Advantages of using FTP**
  - It allows you to transfer multiple files and folders. When the connection is lost then it has the ability to resume the transfer.
  - There is no limitation on the size of the file to be transferred. The browsers allow a transfer of only up to 2 GB.
  - Many FTP clients like FileZilla have the ability to schedule the transfers.
  - The data transfer is faster than HTTP.
  - The items that are to be uploaded or downloaded are added to the 'queue'. The FTP client can add items to the 'queue'.

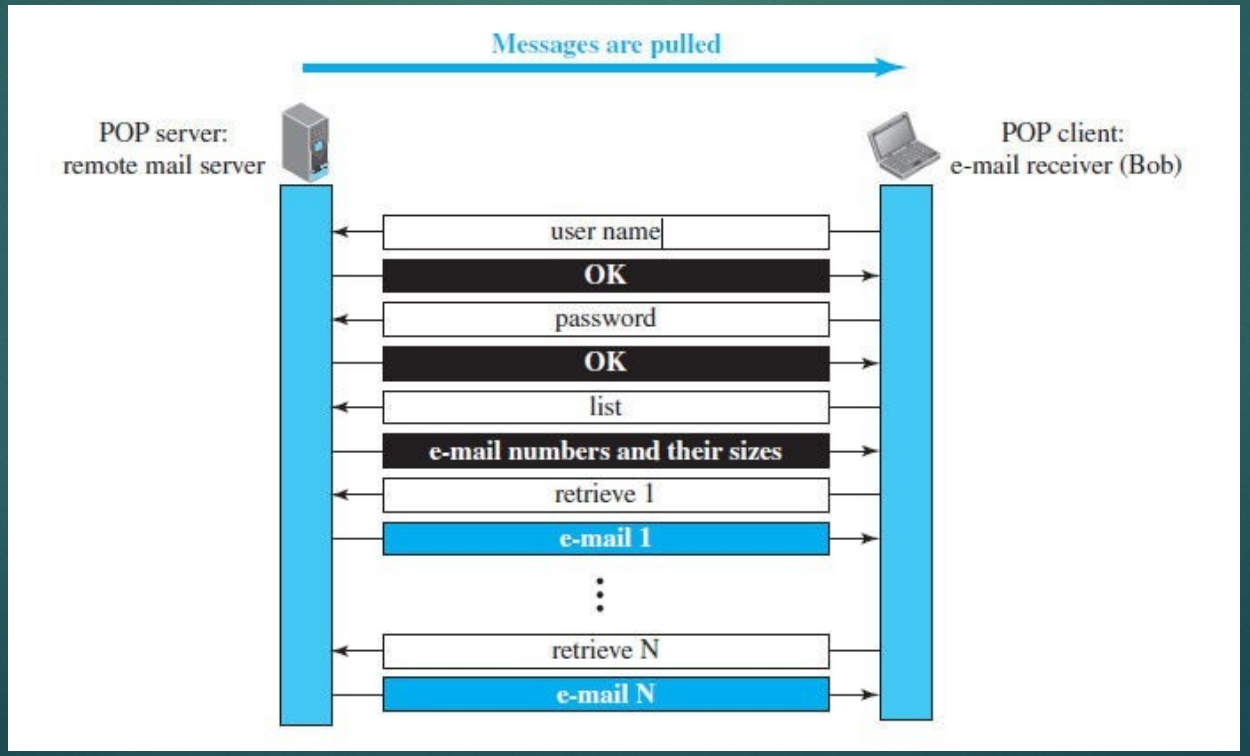
- **Disadvantages of using FTP**
  - **FTP doesn't encrypt the traffic so usernames, passwords**, and other data can easily be read by capturing the data packets because while transferring as they are sent in cleartext. FTP is vulnerable to packet capture and other attacks.

POP3

- **Post Office Protocol, version 3 (POP3)** is simple but limited in functionality. The client POP3 software is installed on the recipient computer; the server POP3 software is installed on the mail server.
- Mail access starts with the client when the user needs to download its e-mail from the mailbox on the mail server.
- The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox.
- The user can then list and retrieve the mail messages, one by one.

POP3

- POP3 has two modes: the delete mode and the keep mode. In the delete mode, the mail is deleted from the mailbox after each retrieval.
- In the keep mode, the mail remains in the mailbox after retrieval



IMAP

- Another mail access protocol is Internet Mail Access Protocol, version 4 (IMAP4).
- IMAP4 is similar to POP3, but it has more features; IMAP4 is more powerful and more complex.
- IMAP4 provides the following functions:
- A user can check the e-mail header prior to downloading.
- A user can search the contents of the e-mail for a specific string of characters prior to downloading.
- A user can partially download e-mail. This is especially useful if bandwidth is limited and the e-mail contains multimedia with high bandwidth requirements.
- A user can create, delete, or rename mailboxes on the mail server.
- A user can create a hierarchy of mailboxes in a folder for e-mail storage.

SNMP

- Simple Network Management Protocol
- It is an **application–layer protocol** for exchanging management information between network devices.
- •It is a part of Transmission Control Protocol / Internet Protocol (TCP/IP) protocol suite.
- •SNMP is one of the widely acceptednetwork protocolsto manage and monitor network elements.
- •Most of the professional–grade network elements come with bundled SNMP agent. These agents have to be enabled and configured to communicate with thenetwork monitoring toolsor network management system (NMS).

SNMP

- SNMP consists of
- SNMP Manager
  - A manager or management system is a separate entity that is responsible to communicate with the SNMP agent implemented network devices.
- SNMP Manager's key functions
  - Queries agents
  - Gets responses from agents
  - Sets variables in agents
  - Acknowledges asynchronous events from agents

SNMP

- Managed devices
  - A managed device or the network element is a part of the network that requires some form of monitoring and management e.g. routers, switches, servers, workstations, printers, UPSs, etc.
- SNMP agent
  - The agent is a program that is packaged within the network element. Enabling the agent allows it to collect the management information database from the device locally and makes it available to the SNMP manager
    - SNMP agent's key functions
    - Collects management information about its local environment
    - Stores and retrieves management information as defined in the MIB.
    - Signals an event to the manager.
    - Acts as a proxy for some non–SNMP manageable network node.
- Management Information Database Otherwise called as Management Information Base (MIB)
  - Every SNMP agent maintains an information database describing the managed device parameters. The SNMP manager uses this database to request the agent for specific information and further translates the information as needed for the Network Management System (NMS). This commonly shared database between the Agent and the Manager is called Management Information Base (MIB)

HTTP

- ▸ Hyper Text Transfer Protocol (HTTP) is used to define how the client-server programs can be written to retrieve web pages from the Web.
- ▸ An HTTP client sends a request; an HTTP server returns a response. The server uses the port number 80; the client uses a temporary port number.
- ▸ HTTP uses the services of TCP, which, as discussed before, is a connection- oriented and reliable protocol.
- ▸ Before any transaction between the client and the server can take place, a connection needs to be established between them. After the transaction, the connection should be terminated.
- ▸ The client and server, however, do not need to worry about errors in messages exchanged or loss of any message, because the TCP is reliable and will take care of this matter

HTTP

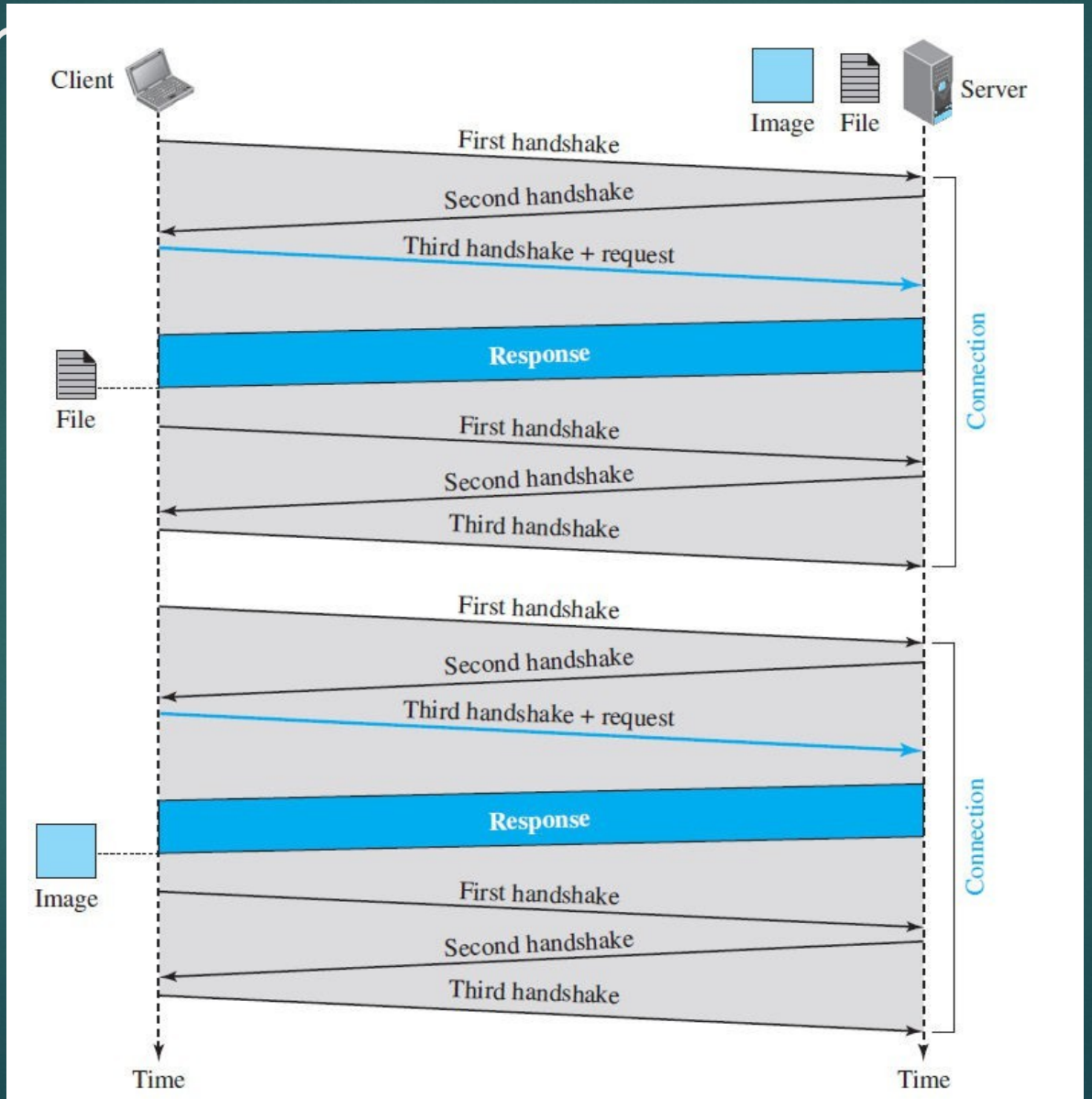
- If the web pages, objects to be retrieved, are located on different servers, we do not have any other choice than to create a new TCP connection for retrieving each object.
- If some of the objects are located on the same server, we have two choices: to retrieve each object using a new TCP connection or to make a TCP connection and retrieve them all.
- The first method is referred to as a *nonpersistent connection,* the second as a *persistent connection*.
  - HTTP, prior to version 1.1, specified *nonpersistent* connections, while *persistent* connections are the default in version 1.1, but it can be changed by the user.

Non-Persistent Connection

- In a **nonpersistent connection**, one TCP connection is made for each request/response.
- The following lists the steps in this strategy:

1. The client opens a TCP connection and sends a request.
2. The server sends the response and closes the connection.
3. The client reads the data until it encounters an end-of-file marker; it then closes the connection.

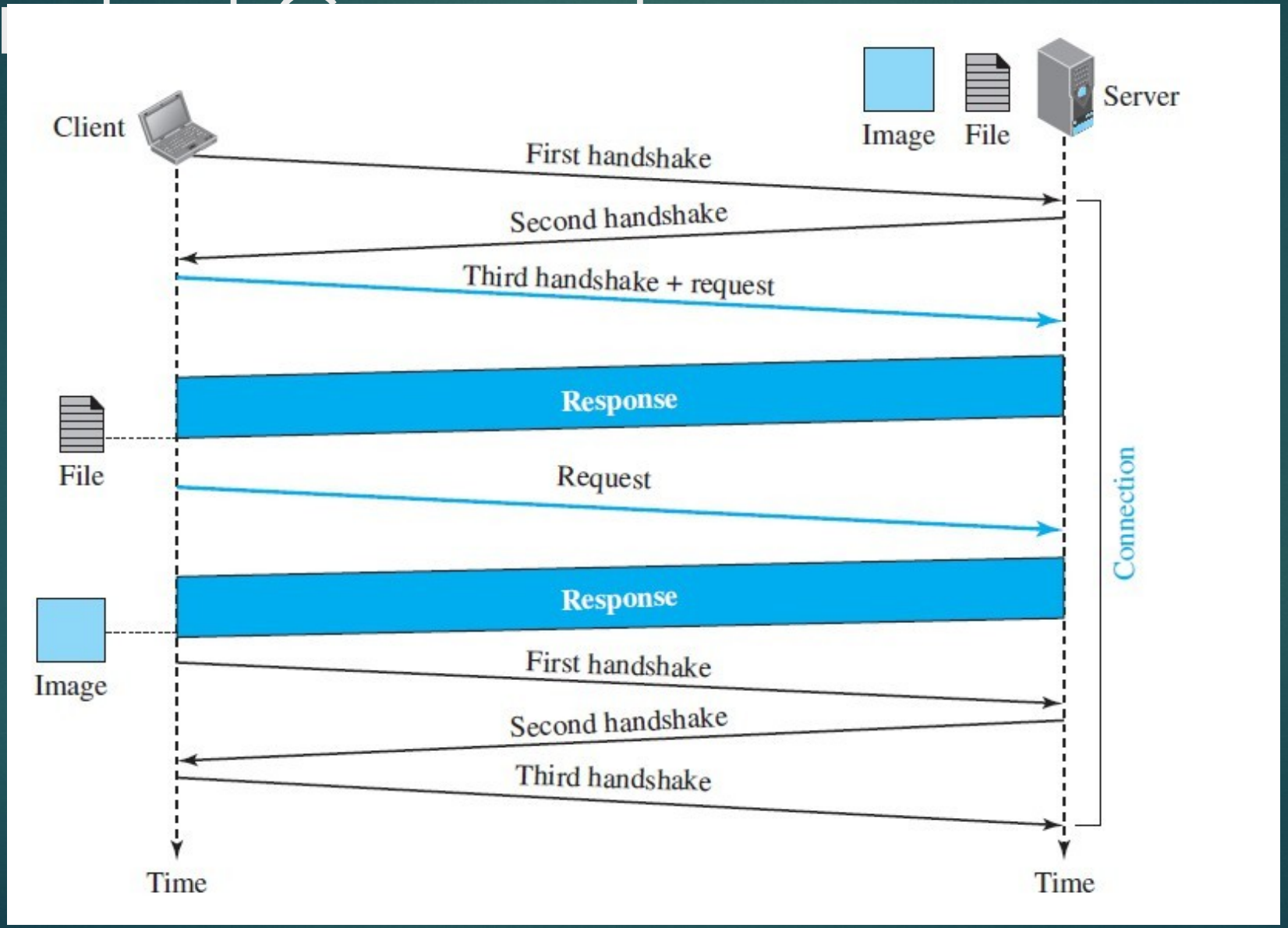
Non-Per


Client
Image
File
Server
First handshake
Second handshake
Third handshake + request
Response
File
First handshake
Second handshake
Third handshake
Connection
First handshake
Second handshake
Third handshake + request
Response
Image
First handshake
Second handshake
Third handshake
Connection
Time
Time

Persistent Connection

- HTTP version 1.1 specifies a **persistent connection** by default. In a persistent connection, the server leaves the connection open for more requests after sending a response.
- The server can close the connection at the request of a client or if a time-out has been reached.
- The sender usually sends the length of the data with each response.
- There are some occasions when the sender does not know the length of the data. This is the case when a document is created dynamically or actively.
- In these cases, the server informs the client that the length is not known and closes the connection after sending the data so the client knows that the end of the data has been reached. Time and resources are saved using persistent connections

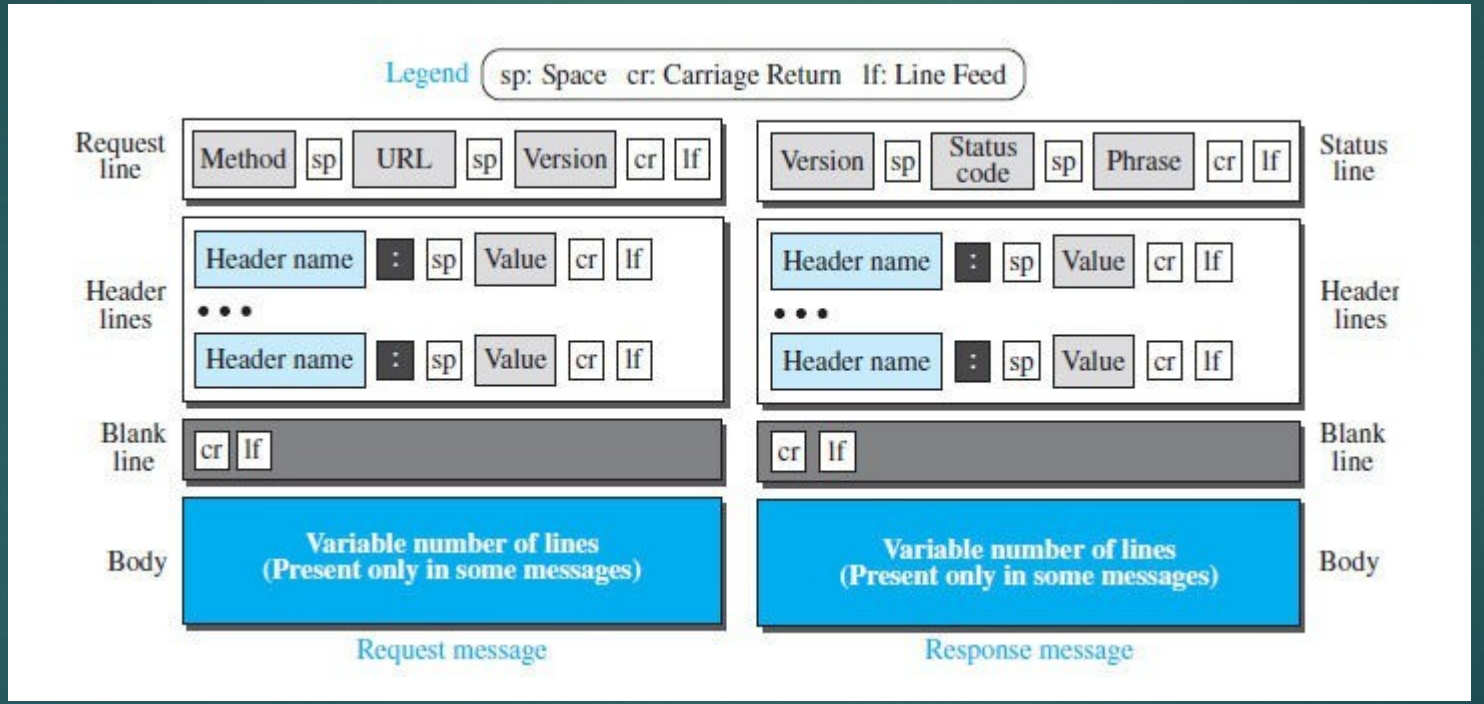
Pe


Client
Image
File
Server
First handshake
Second handshake
Third handshake + request
Response
File
Request
Connection
Response
Image
First handshake
Second handshake
Third handshake
Time
Time

Message Format

- The HTTP protocol defines the format of the request and response messages, as shown in Figure
- Each message is made of four sections.
- The first section in the request message is called the request line; the first section in the response message is called the status line.
- The other three sections have the same names in the request and response messages.
- similarities between these sections are only in the names; they may have different contents.

Message Format



Request Message

- the first line in a request message is called a request line. There are three fields in this line separated by one space and terminated by two characters (carriage return and line feed)
- The fields are called method, URL, and version.
- The method field defines the request types. In version 1.1 of HTTP, several methods are defined, as shown in Table Below
- The second field, URL, it defines the address and name of the corresponding web page.
- The third field, version, gives the version of the protocol; the most current version of HTTP is 1.1.

Method	*Action*
GET	Requests a document from the server
HEAD	Requests information about a document but not the document itself
PUT	Sends a document from the client to the server
POST	Sends some information from the client to the server
TRACE	Echoes the incoming request
DELETE	Removes the web page
CONNECT	Reserved
OPTIONS	Inquires about available options

Request Message

- After the request line, we can have zero or more request header lines. Each header line sends additional information from the client to the server. For example, the client can request that the document be sent in a special format. Each header line has a header name, a colon, a space, and a header value
- The body can be present in a request message. Usually, it contains the comment to be sent or the file to be published on the website when the method is PUT or POST.

Response Message

- A response message consists of a status line, header lines, a blank line, and sometimes a body. The first line in a response message is called the *status line*.
- There are three fields in this line separated by spaces and terminated by a carriage return and line feed. The first field defines the version of HTTP protocol, currently 1.1.
- The status code field defines the status of the request. It consists of three digits.
- Whereas the codes in the 100 range are only informational, the codes in the 200 range indicate a successful request. The codes in the 300 range redirect the client to another URL, and the codes in the 400 range indicate an error at the client site. Finally, the codes in the 500 range indicate an error at the server site. The status phrase explains the status code in text form.

Response Message

- After the status line, we can have zero or more *response header* lines. Each header line sends additional information from the server to the client. For example, the sender can send extra information about the document. Each header line has a header name, a colon, a space, and a header value
- The body contains the document to be sent from the server to the client. The body is present unless the response is an error message